Project Proposal for 15-400

Anlun Xu

November 2018

1 Project Title

Effect System in Wyvern

2 Project Webpage

http://anlunx.github.io

3 Project Description

In this project, I will be working with Professor Jonathan Aldrich and Professor Alex Potanin.

This project is about improving the effect system in Wyvern programming language to make it more user-friendly. First of all, the Wyvern is a new language designed by Professor Jonathan Aldrich and Professor Alex Potanin to help programmers be highly productive when writing high-assurance applications. The role of effect system in Wyvern is to help programmers to reason about the effect of the code. For example, programmers can know if a program can write to a file or use the network by looking at the code. Effect system prevent code that has undesired effect from being executed, therefore ensuring the security of the program. However, the issue with the current implementation of the effect system is that it requires programmers to explicitly annotate the effects for every program. This requirement causes a large overhead for programmers in terms of the number of annotations in program.

Therefore, this project aims to reduce the overhead of effect annotations by implementing a capability-based reasoning about effect in Wyvern. The original idea is from the Craig *et al.* (1), which is to bound the effects of an expression based on the capabilities to which it has access. The paper built the formal definitions of the new system on a typed lambda calculus. In this project, I will adapt this idea to the context of Wyvern, and design the implementation of the new Effect System.

Since we can bound the effects using capabilities, we can allow effect unannotated code in our effect system. So the amount of annotations required is reduced. Therefore, the impact of this project would be that the new effect system would be easier to use.

4 Project Goals

In addition to implementing the effect system, I would also conduct some case studies and a human-subject experiment to show the improvements on the effect system.

1. 75% Goal:
   Complete the Effect system, case studies and human-subject experiment.

2. 100% Goal:
   Complete the effect system, case studies and human-subject experiment.
   Add bounded polymorphic effects
   Enforce separation of effect-annotated and effect-unannotated code

3. 125% Goal:
   Complete the goals above, and add new type theory/new language features to Wyvern

5 Milestones

1st Technical Milestone for 15-300:
Adapt the checking rules to concrete Wyvern language

Bi-weekly milestones:

1. February 1st: Do case studies and measure the number of effect annotations (with/without quantification lifting)
2. February 15th: Prepare for Human subject experiments: make plans for the experiment and submit a proposal to IRB
3. March 1st: Prepare for Human subject experiments: design programming tasks that are used in experiment.
4. March 22nd: Finish Human subject experiments: comparing productivity of programmers on some tasks
5. April 5th: Implement bounded effect polymorphism
6. April 19th: Do more case studies based on bounded effect polymorphism

7. May 3nd: Enforce separation of effect-annotated and effect-unannotated code

6 Literature Search

Papers I have read so far:

1. Aaron Craig, Alex Potanin, Lindsay Groves, and Jonathan Aldrich, Capabilities: Effects for Free(1)

   This paper proposes the theory of bounding effects using capabilities in typed lambda calculus

2. Lubin, J. (2018) Approximating Polymorphic Effects with Capabilities.SPLASH 2018 Student Research Competition

   This paper propose a method to handle the mixing of effect-annotated and effect-unannotated code in Wyvern.

7 Resources Needed

I will use funds from my advisor to pay human subjects in human subjects experiments. I have all of the resources that I need to conduct this study.

References

[1] Aaron Craig, Alex Potanin, Lindsay Groves, and Jonathan Aldrich *Capabilities: Effects for Free.*